

The MIXMAX random number generator

April 2, 2014

Konstantin G. Savvidy¹

Department of Physics

and

Center for Transcriptional Medicine,
Nanjing University, Nanjing, China

Abstract

In this note, we give a practical solution to the problem of determining the maximal period of matrix generators of pseudo-random numbers which are based on an integer-valued unimodular matrix of size $N \times N$ known as MIXMAX and arithmetic defined on a Galois field $\text{GF}[p]$ with large prime modulus p . The existing theory of Galois finite fields is adapted to the present case, and necessary and sufficient condition to attain the maximum period is formulated. Three efficient algorithms are presented. First, allowing to compute the multiplication by the MIXMAX matrix with $O(N)$ operations. Second, to recursively compute the characteristic polynomial with $O(N^2)$ operations, and third, to apply skips of large number of steps S to the sequence in $O(N^2 \log(S))$ operations. It is demonstrated that the dynamical properties of this generator dramatically improve with the size of the matrix N , as compared to the classes of generators based on sparse matrices and/or sparse characteristic polynomials. Finally, we present the implementation details of the generator and the results of rigorous statistical testing.

¹ksavvidis(AT)gmail.com

1 Introduction

In [5] it was proposed that k-mixing systems of Kolmogorov [1, 2, 3] may serve as a suitable random number generator. The particular system chosen was the one realizing linear automorphisms of the unit hypercube in \mathbb{R}^N :

$$u_i(t+1) = \sum_{j=1}^N A_{ij} u_j(t) \bmod 1 \quad (1)$$

where $u \in [0, 1)$. For the purposes of generating pseudo-random numbers with this method, one chooses the initial vector $u(0)$, called the “seed”, with at least one non-zero component.

The entries of the matrix are integers: $A_{ij} \in \mathbb{Z}$ and subject to the following two conditions [1, 2, 3, 5] on the defining matrix A :

- $\det A = 1$
- the eigenvalues λ_k of A must not lie on the unit circle, $|\lambda_k| \neq 1$ for all $k = 1 \dots N$.

The first condition assures that the map defines a volume preserving automorphism. When the automorphism is viewed as defining a dynamical system with discrete time $t \in \mathbb{Z}$, then the second condition assures that nearby trajectories diverge exponentially. There exists an everywhere dense, but discrete set of periodic trajectories all of which are unstable, and whose number as a function of the period τ asymptotically goes like $e^{h\tau}/\tau$, where h is the Kolmogorov entropy of the system [1, 4]:

$$h = \sum_{k: |\lambda_k| > 1} \log |\lambda_k|$$

The auto-correlation decay time τ_0 is related to the entropy as $\tau_0 \leq 1/h$. In this way, a connection is made between the chaotic dynamics of the system, its set of periodic trajectories and the entropy. In particular, the auto-correlation time is directly related to the empirical notion of randomness of the sequence. The second condition above states that none of the eigenvalues should lie precisely on the unit circle, and moreover the formula for the entropy demonstrates that it is desirable that most of the eigenvalues of the matrix should lie as far as possible away from it.

A particular matrix chosen in [6] was defined for all $N \geq 3$:

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 2 & 1 & 1 & \dots & 1 & 1 \\ 1 & 3+s & 2 & 1 & \dots & 1 & 1 \\ 1 & 4 & 3 & 2 & \dots & 1 & 1 \\ & & & & \dots & & \\ 1 & N & N-1 & N-2 & \dots & 3 & 2 \end{pmatrix} \quad (2)$$

The MIXMAX matrix contains integer numbers and is defined recursively, since the matrix of size $N + 1$ contains in it the matrix for size N . The only variable entry in the matrix is $A_{32} = 3 + s$ where s is some small “magic” integer, in many cases $s = -1$ or $s = 0$. For those N for which one or more eigenvalues lie on the unit circle for some s , one can choose another s .

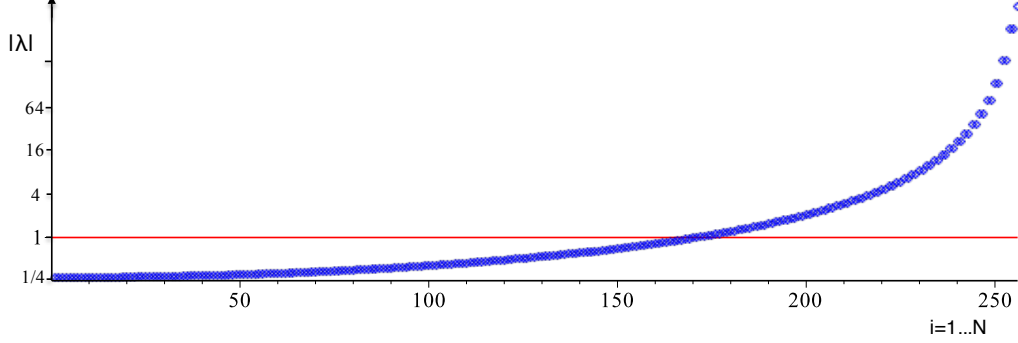


Figure 1: The absolute value of the eigenvalues of the MIXMAX matrix for $N=256$ (logarithmic scale). Almost all of the eigenvalues are far away from the unit circle.

The eigenvalues of the MIXMAX matrix are widely dispersed for all N , see Figure 1. Thus, the spectrum of this system is multi-scale, with trajectories exhibiting exponential instabilities on all time-scales [5].

Empirical evidence suggests that the largest eigenvalue appears to grow at least linearly with N , but we have been unable to obtain a strict bound on it. However, Kolmogorov’s entropy can be more conveniently calculated using the small eigenvalues as follows. Since the product of all of the eigenvalues is equal to the determinant,

$$\prod_k \lambda_k = \prod_k |\lambda_k| = 1, \quad \text{and} \quad \sum_k \log |\lambda_k| = 0,$$

the entropy can be calculated equally well using the eigenvalues which are less than one by absolute value:

$$h = - \sum_{k: |\lambda_k| < 1} \log |\lambda_k|$$

i.e. the entropy is equal to the area under the upper branch of the curve in Fig. 1 and also is equal to the area under the lower branch of the curve, taking into account that the vertical axis is already set to the logarithmic scale.

None of the eigenvalues of the Matrix A is smaller by absolute value than $1/4$, regardless of N , and a large number of them tend to cluster just above $1/4$. Therefore the Kolmogorov entropy can be strictly bound from above by bounding the area under the lower branch of the curve as follows:

$$h < -N \log |\lambda_{\min}| < N \log(4).$$

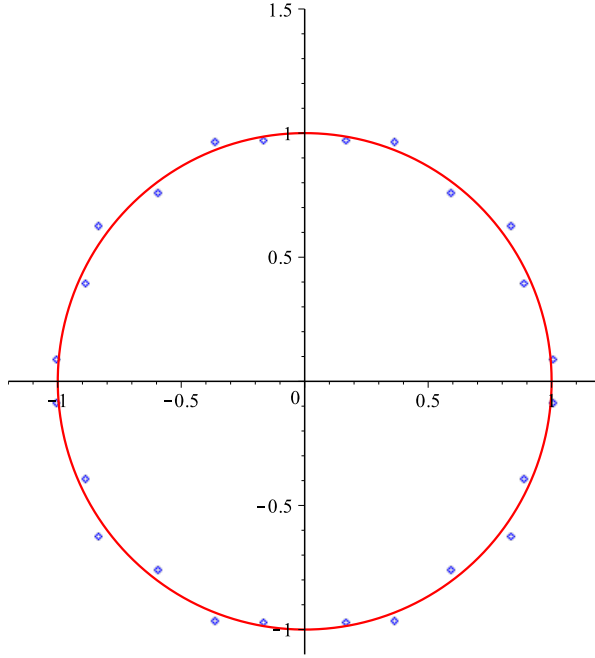


Figure 2: Eigenvalues of the defining matrix of the RCARRY generator [13] all lie close to the unit circle. The eigenvalues closest to the circle have $|\lambda| \approx 1.0085$, the farthest $|\lambda| \approx 1.043$. In order to overcome the strong correlations exhibited by RCARRY as a result of this property, Lüscher proposed to skip a predefined number of steps in the sequence [14].

We have also obtained an heuristic formula for the entropy which is more precise. The actual number of eigenvalues lesser than one by absolute value is asymptotically $2/3 N$, and logarithm of their value is observed to lie close to a parabola:

$$\log(\lambda_k) \lesssim \log(4) \left(-1 + \left(\frac{3}{2N} \right)^2 k^2 \right) \quad \text{for } k = 1 \dots \frac{2N}{3}. \quad (3)$$

Adding these up, we get an approximate asymptotic formula, which also satisfies the strict bound above:

$$N \log(4) > h \gtrsim 4/9 N \log(4)$$

This estimate appears to be reasonable, i.e. for $N = 256$ the actual value is $h \simeq 164.4$ versus our estimate of $h \gtrsim 157.7$.

Figure 2 demonstrates graphically that some of the other popular generators do not satisfy the second requirement for randomness. In the case of RCARRY [13] which is a slight modification of a Fibonacci-like recurrence modulo 2^{24} , this point has been made before by Lüscher [14], and its failure was related to the weak mixing properties of its underlying matrix. Unfortunately, the Mersenne

Twister (MT) [15] has not been studied from this point of view. Our preliminary investigation indicates that the real eigenvalues of its characteristic polynomial are distributed very close to the unit circle, with the largest being less than $|\lambda_{max}| < 1.0019$, see Fig. 3. In this sense, the underlying dynamical system of the Mersenne Twister has one order of magnitude less entropy than RCARRY. This is related to the singular flaw acknowledged by the authors of the Mersenne Twister, which is that MT has a very long recovery time when it is seeded by a vector containing mostly zeros: the output is observed to be non-random even after outputting a million values. In this instance, the divergence of some trajectory is observed away from the origin. In fact, from the dynamical system point of view *this is not merely a manifestation of an unlucky initialization*, since any two nearby trajectories of the MT diverge very slowly and this is ultimately what causes the failure of MT in the statistical tests.

The total entropy of the MT system is approximately $h \approx 4.8$. On the basis of the known behavior of RCARRY and our investigation of the MIXMAX, it appears that an entropy of at least $h \gtrsim 50$ is required for a generator to have a sufficiently random trajectory. In light of this, it is perhaps not surprising that the Mersenne Twister fails many tests in its pure form, and still fails some tests even when some additional tempering of the output is applied.

One can even conjecture that the sequence produced by the Mersenne Twister would equally benefit from the method of skipping proposed by Lüscher. However, the required amount of skipping would likely be prohibitively expensive. Moreover, it is clear that the recurrences based on primitive trinomials of even higher order such as those found in [17] would exhibit even longer correlations.

2 Discrete case

In a typical computer implementation, the recursion (1) can be used to generate uniform random variables on the unit interval directly in floating point hardware. However, since actual floating point hardware has finite precision the sequence will tend to lie on a rational sublattice, with components of the vector x_i being multiples of 2^{-b} where b is the number of bits available for the mantissa of the floating point unit, typically $b = 53$ on current computers. There are at least two drawbacks to this scheme. First, the period of the generator so realized will strongly depend on the initial seed. Second, the operation of truncation of the integer part which is indicated in eq. (1) may not be particularly efficient.

We now consider this from another point of view [11]. If the initial vector happens to have rational components $u_i = m_i/n_i$, where m and n are natural numbers, then all subsequent vectors will remain on the rational sub-lattice $u_i = a_i/p$ where p is the least common multiple of the n_i . In this case it is convenient to represent u_i by its numerator a_i in computer memory, or even better, to simply define

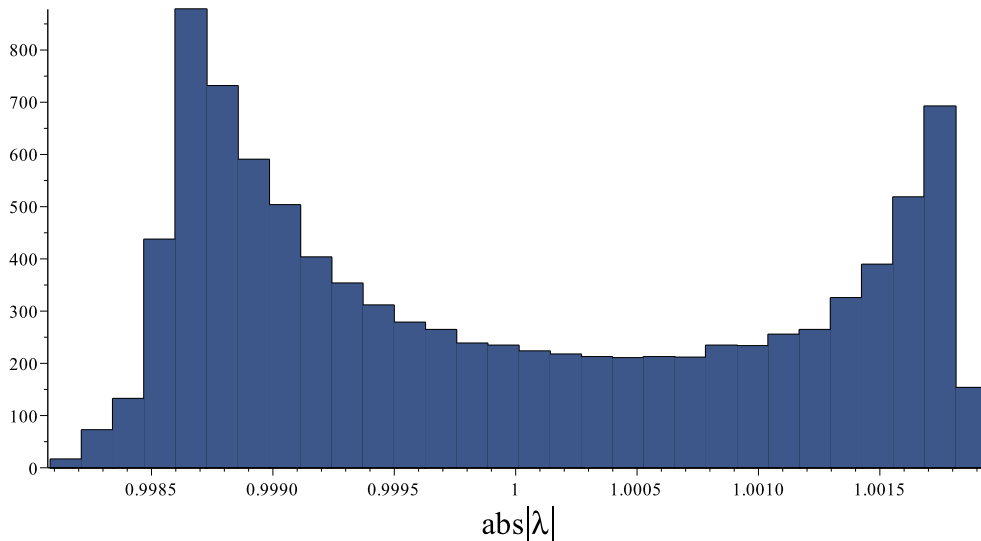


Figure 3: Eigenvalues of the defining matrix of the Mersenne Twister generator [13] all lie close to the unit circle. The smallest (by absolute value) eigenvalue has $|\lambda| \approx 0.9982$, the largest $|\lambda| \approx 1.0019$.

the same recursion in terms of a_i :

$$a'_i = \sum_{j=1}^N A_{ij} a_j \bmod p \quad (4)$$

In the rest of the paper we shall not indicate the modular operations explicitly, and use the equal sign in the sense of equivalence modulo p .

Random number generators of this general form, without reference to the underlying automorphism of the torus and the dynamical systems theory were proposed by Tahmi [7] and Niki [8] and were extensively studied by Grothe, Niederreiter and others [9, 11, 10, 12]. These authors established the connection to the theory of finite extended Galois fields, and obtained results regarding the period of the recursion when the modulus p is a prime.

On a given computer architecture, p could be chosen as the largest ordinary or Mersenne prime lesser than the maximum unsigned integer. A distinct class of recursions results if $p = 2$, in which case the GF[2] arithmetic may be naturally realized on the available computer architectures via the bitwise XOR operation.

We now summarise the known mathematical facts about such linear matrix recursions, modulo a prime number p . The sequence of vectors $a(t)$ generated by the recursion is necessarily periodic, and the period cannot exceed the cardinality of the nonzero elements of the finite vector field, which is equal to $p^N - 1$. If, and only if, the characteristic polynomial $P(x)$ of the matrix A , $P(x) = (-1)^N \det |A - x \mathbb{I}|$ is primitive in the extended Galois finite field $GF[p^N]$, then the period of the recursion attains its

maximal value, $p^N - 1$, independent of the seed. The necessary and sufficient conditions for this are well known [12]. One of the necessary conditions is the following:

- the free term of the polynomial $p_0 = P(x)|_{x=0}$, also equal to the determinant of the matrix $p_0 = \det A$, is a primitive element of the Galois field $\text{GF}[p]$

The determinant of the MIXMAX matrix is equal to one, since it is one of the conditions for Kolmogorov k-mixing to occur, and for the mapping to be an automorphism. Therefore, as it was noted in [11], the recursion defined by (4) cannot attain the maximal period (unless $p = 2$). In the next section we investigate the maximal possible period of the sequence (4) for $p \neq 2$, and generalize the notion of primitive polynomials to the required case.

3 Maximal period sequence

A key insight which allows to determine the maximal period of the sequence defined by (4) is that, in the case that the characteristic polynomial primitive,

$$A^q = p_0 \mathbb{I} \quad \text{for } q = \frac{p^N - 1}{p - 1} ,$$

where as before p_0 is the free term of the characteristic polynomial of A and is equal to the determinant of A . Therefore, the powers of the matrix A which are multiples of q are diagonal:

$$A^{mq} = (A^q)^m = p_0^m \mathbb{I}$$

Since the MIXMAX matrix has $p_0 = 1$, therefore the maximum possible period is equal to q . We define two necessary and sufficient conditions for the period τ of the unimodular matrix recursion to attain its maximum possible value and be equal to q :

1. $A^q = \mathbb{I}$
2. $A^{q/r} \neq \mathbb{I}$ for any r which is a prime divisor of q

In practice, it is convenient to calculate the modular exponentiation of a matrix using the theorem of Hamilton and Cayley. Since $P(A) = 0$, we can always reduce some high power of the matrix in terms of powers up to $N - 1$:

$$A^m = e_{N-1}A^{N-1} + \dots + e_2A^2 + e_1A + e_0 , \tag{5}$$

where the coefficients can also be obtained by polynomial algebra:

$$E(x) = x^m \bmod P(x) = e_{N-1}x^{N-1} + \dots + e_2x^2 + e_1x + e_0 . \tag{6}$$

For the first condition to be satisfied, it is necessary and sufficient that the characteristic polynomial is irreducible: $P(x) \bmod Q(x) \neq 0$ for any polynomial $Q(x) \neq 0$. The second condition can be checked directly, by computing $x^{q/r} \bmod P(x)$ for all r which are prime divisors of q . If it is violated for some r , then the actual period of the sequence is equal to the q divided by the least common multiple of any such r . If both conditions are satisfied, then we may call the corresponding characteristic polynomials “quasi-primitive”. As it should be obvious from the definition, the only condition which is relaxed compared to the ordinary, primitive polynomials, is that the free term of the characteristic polynomial p_0 is not required to be the primitive element of the base field $\text{GF}[p]$. In the particular case of interest to us, the matrix A in (2) has $p_0 = \det A = 1$.

From here, it follows that the period of the sequence is equal to q , and is independent of the seed. Moreover, there are precisely $p - 1$ disjoint sequences which together fill up the entire space of states: $q(p - 1) = p^N - 1$. It appears to be a difficult mathematical problem to decide whether two given vectors belong to the same or different sequence. However, one may jump from one sequence to another non-overlapping sequence by multiplying the initial vector $a(0)$ or indeed any subsequent vector $a(t)$ by a number c , subject to some conditions on c . We will be able to lift this minor restriction, and make a more complete description of the entire set of disjoint trajectories in the next section.

All of the above preceding mathematical statements can be justified on the basis of the theory of finite fields, specifically most of the necessary proofs can be obtained by applying the method developed in the Lemma 3.17 in the book by Lidl and Niederreiter [12].

Furthermore, it is possible to carry out the check in the second condition only if the full integer factorization of q is available. In the contrary case, we may suppose that q is only partially factorized:

$$q = r_1^{\alpha_1} \dots r_k^{\alpha_k} c_1 \dots c_m, \quad (7)$$

where r_i are the prime divisors, α their multiplicities and c_i are the composite co-factors of q . Further, suppose that it is known that all of c_i have no divisors smaller than some common lower bound u . Assuming that the second condition holds for all of r_i and c_i , we can obtain the strict bound on the period of the sequence:

$$q \geq \tau > r_1^{\alpha_1} \dots r_k^{\alpha_k} u^m$$

If u , the lower bound on the unknown divisors is sufficiently high, then one can state with very high certainty that nevertheless the period τ is in fact equal to q , with probability that goes asymptotically as $1 - u^{-1} \rightarrow 1$.

As a practical matter, it is contemplated to use $p = 2^{61} - 1$, the largest Mersenne number that fits into an unsigned integer on current 64-bit computer architectures. Fortunately, this is enough to produce double-precision floating point values which are random down to their lowest bit.

The complete integer factorization of $q = \frac{p^N-1}{p-1}$ for some N is available. Typically, it is easier to factorize for some N , and then reuse the same factors for $N = kN$ for some small k . The greatest benefit from these algebraic factorizations results if N is a product of some small primes, for example a primorial. It is not difficult to find all of the small divisors for any N by means of the Elliptic Curve Method (ECM), this is useful for reasons explained above.

We have made searches for irreducible polynomials for various values of N and the “magic” number s , which are summarized in Table 1. The period of the sequence for all of the given parameters is astronomical, and cannot be exhausted even if the number of parallel instances of the generator is itself astronomical. Nevertheless, the generator fails statistical tests for small N . Therefore, the period by itself cannot be considered a measure of quality of a generator. On the other hand, the entropy and the empirical randomness of the generator gets uniformly better with N . For all $N > 64$ and $h \gtrsim 50$ which we have tested, the generator passes all tests.

We recommend to choose the value of N based on the problem at hand. From the dynamical systems point of view, Monte Carlo simulation of a Markov Chain (MCMC) should be done with a random number generator whose auto-correlation time $\tau_0 = 1/h$ is much smaller than the auto-correlation time of the Markov chain. This is easily satisfied by choosing a sufficiently large size N of the generator matrix. On the other hand, difficulties also arise when simulating a system with very long auto-correlation time, for example the Ising model with the Metropolis method near the critical point. In this case, and other cases where the effective dimension of the Monte-Carlo integration is large, one should choose N which is larger than this effective dimension. Usually, this requirement is stated in terms of the equidistribution of the sequence. No generator can guarantee equidistribution in a dimension larger than the dimension of its internal state [18].

For $p = 2$, q may happen to be a prime, called Mersenne prime. In the past, useable random number generators could be constructed on the basis of some of the available Mersenne primes, such as $N = 19937$. In many of these cases, a search is made for a primitive trinomial in $\text{GF}[2]$ of order N , and a corresponding Fibonacci recursion is used to generate random bits (the GFSR family). Otherwise, as in the case of the Mersenne Twister, a sparse matrix is constructed of the almost-banded form whose characteristic polynomial is proved to be irreducible for some values of the magic entries. Since q is prime, the second condition is satisfied automatically, and therefore the characteristic polynomial is truly primitive in these cases.

4 The case of the prime period

The contents of the previous section can be given an elegant extension by the following observation:

if N is prime, then q may happen to be a prime number as well and therefore the period τ is

Size N	Magic s	Entropy (lower bound)	Period τ/q	$\approx \log_{10}(q)$	q is fully factored	BigCrush
10	-1	6.2	1/4	165	Yes	33
16	6	9.9	1/32	275	Yes	> 13
40	1	24.6	1/4	716	Yes	3
44	0	27.1	1/4	789	No	4
60	4	37.0	1	1083	Yes	2
64	6	39.4	1/8	1156	No	1 (?)
88	1	54.2	1/2	1597	No	Pass
256	-1	157.7	1	4682	No	Pass
508	5	313.0	1	9309	No	Pass
720	1	443.6	1	13202	No	Pass
1000	0	616.1	1/20	18344	No	Pass
1260	15	776.3	1/2	23118	No	Pass
3150	-11	1940.8	1/12	57824	No	Pass

Table 1: Table of properties of generators for different matrix size N and special magic value s . For each N that we investigated, the period τ is given as a fraction of $q = (p^N - 1)/(p - 1)$. For cases where the full integer factorization of q is known, unconditional guarantee can be given about the period of the sequence. In all cases the characteristic polynomial was proved to be irreducible by Pari/GP [19]. The last column indicates whether the generator for that N and special value s passes the BigCrush suite of tests, and if not how many tests are failed. The case of $N = 60$ uses a doubly special matrix which has two entries modified: $a_{32} = a_{54} = 3 + s$. It is seen that the generator gets uniformly better with N until it passes all tests. The most discriminative test for this family of generators appears to be the classic Gap test. On this test alone, the improvement with N is also evident, with progressively better p-values as N is increased, e.g. for $N=64$ the value of $\chi^2 \approx 372$ for 232 degrees of freedom with $\chi^2/dof \approx 1.6$ indicates only a marginal failure. For all $N > 64$ which we have tested, the generator passes all tests.

guaranteed to be equal to q , $\tau = q$, if and only if the characteristic polynomial is irreducible.

There are no algebraic factors of q if N is an odd prime. There are no trivial factors if N is co-prime with $p - 1$. Other than this, naive considerations indicate that the *a priori* probability that some particular q is a prime is finite, and goes like $\frac{1}{N \log(p)}$. For $p = 2$, if $2^N - 1$ happens to be a prime it is called a Mersenne prime. Primes of the form $\frac{p^N - 1}{p - 1}$ for some prime N and a small prime base p are called “repunit” primes, and it is a natural extension of the notion of Mersenne primes to $p \neq 2$. Our q are “repunit” numbers in base p , since $q = 1 + p + p^2 + \dots + p^{N-1}$ (a repunit number has all digits equal to one in some base, here p). However, the bases we use are very large, so it is somewhat unnatural to call it “repunit”, a more appropriate name might be “affine-Mersenne” prime.

The period q may happen to be a prime also when p itself is an ordinary Mersenne prime, but unfortunately, for $p = 2^{61} - 1$ we did not find any such N for which q is a prime number. Not to be discouraged, we have made a non-exhaustive search for some other primes p , chosen for convenience to be just short of 2^{62} or 2^{63} . The search has yielded the combinations of p , N and s for which N and q are prime, and the characteristic polynomial of the MIXMAX matrix (for such s) is irreducible. The second condition of the previous section is then trivially fulfilled and therefore the characteristic polynomial is quasi-primitive. The results are presented in Table 2.

In all of the cases in Table 2 we have $\gcd(q, p - 1) = 1$. Compared to most of the cases considered in the previous section, in Table 1, this gives the additional benefit: the matrix A^m is not diagonal for any power of m less than q :

$$A^m \neq c\mathbb{I} \text{ for any } c = 1 \dots (p - 1) \text{ and } m < q \quad (8)$$

unlike many of those in Table 1. For example, whenever the period τ is even, we have

$$A^{\tau/2} = -\mathbb{I} = (p - 1)\mathbb{I} \quad (9)$$

or, more generally

$$A^{\tau/r} = \sqrt[r]{1}\mathbb{I} \quad (10)$$

for any r which is a common divisor of q and $p - 1$ (since $\sqrt[r]{1} \in GF(p)$ whenever $r \mid (p - 1)$). Obviously, this annoyance does not occur when q is a prime number.

In practice, this means that one can jump from any one disjoint trajectory of the generator to another by multiplying the state vector by a number in the range $2 \dots (p - 1)$.

5 Efficient Implementation

In this section, we give for completeness the formulae which allow the efficient implementation of the generator in actual computer hardware.

Modulus p	Size N	Magic s	Period $\approx \log_{10}(q)$	BigCrush
4611686018427341489	17	0	298	14
4611686018427246217	19	0	335	7
4611686018427365419	23	0	410	6
4611686018427297023	31	0	559	3
4611686018427370139	37	-1	671	2
4611686018427317411	43	-1	783	1
4611686018427335557	47	-1	858	1
4611686018427262387	53	1	970	1
4611686018427084347	59	3	1082	1
4611686018426896543	61	0	1119	1
4611686018427033523	67	2	1231	1
4611686018427208187	71	4	1306	1
4611686018426592983	127	0	2351	Pass
9223372036853751941	139	0	2617	Pass
9223372036854661783	257	-4	4855	Pass

Table 2: Table of properties of generators for different matrix size N and special magic value s . For all of these generators, the period τ is equal to q , $\tau = q = (p^N - 1)/(p - 1)$. Since q is prime for all combinations of p and N in this Table, unconditional guarantee can be given about the period of the sequence. In all cases the characteristic polynomial was proved to be irreducible by Pari/GP [19]. The last column indicates whether the generator for that N and special value s passes the BigCrush suite of tests, and if not how many tests are failed. It is seen that the generator gets uniformly better with N until it passes all tests.

First, we present the formula which allows the efficient calculation of the recursion. Given the vector a with components a_i , $i = 1 \dots N$, a vector of partial sums b is formed according to

$$\begin{aligned} b_1 &= 0 , \\ b_i &= b_{i-1} + a_i \quad \text{for } i = 2 \dots N . \end{aligned} \tag{11}$$

Then, the new vector is calculated:

$$\begin{aligned} a'_1 &\leftarrow a_1 + b_N , \\ a'_i &\leftarrow a_{i-1} + b_i \quad \text{for } i = 2 \dots N . \end{aligned} \tag{12}$$

Finally, the correction due to the magic value is applied

$$a'_3 \leftarrow a'_3 + s a_2$$

It is obvious that the above is a unimodular linear transformation, but it is somewhat less trivial to see that (11) and (12) indeed implement the multiplication by the MIXMAX matrix of (2).

Next, we present without proof the recursive formulae which allow the efficient calculation of the characteristic polynomial $P_N(x)$ of the MIXMAX matrix:

$$\begin{aligned} M_0 &= 1 , \\ M_1 &= 2x , \\ M_2 &= 3x^2 + x , \\ M_j &= (2x) M_{j-1} + (1-x)x M_{j-2} \end{aligned} \tag{13}$$

and finally

$$P_N(x) = -x \left[(2x + s) M_{N-3} + (1-x)(x + s) M_{N-4} \right] + (x-1)^N \tag{14}$$

Finally, the formulae (6) and (5) allow the realization of skipping. Fast application of the skipping matrix A^m to the state vector $a(t)$ is possible without actually computing or storing it. Instead, one pre-computes and stores the coefficients e_i , $i = 0 \dots (N-1)$ of the polynomial $E(x)$ defined in (6), for example for a modest set of m which are powers of two up to 1024: $m = 2^j$, $j = 0 \dots 1024$.

According to (5),

$$a(t+m) = A^m . a(t) = E(A) . a(t) = \sum_{i=0}^{N-1} e_i A^i . a(t) = \sum_{i=0}^{N-1} e_i a(t+i) \tag{15}$$

Furthermore, skipping is additive since the different powers of the matrix A commute. Therefore, it is possible to apply skipping by an arbitrary integer $S < 2^{1024}$ by decomposing it as a sum of powers of two, $S = \sum 2^j b_j$, where b_j is the j th bit of S . Then one simply scans over j , and applies the above formula recursively for all nonzero bits of S with $m = 2^j$.

6 Conclusion and Acknowledgements

The MIXMAX random number generator is currently made available by the author in a portable implementation in the C language at hepforge.org [20]. We urge the reader to consult the README file included in the distribution at hepforge.org for details on the use of the generator. Also, an experimental implementation intended as part of a development version of the C++ ROOT library from CERN exists at [21].

The generator outputs in double precision natively, with all 53 bits random. The speed of the generator compares favorably to other generators currently available: in our tests it is significantly faster than RANLUX in 24-bit and 48-bit precision and still somewhat faster than the 32-bit precision Mersenne Twister. Also, among the generators with a large state space it is, to our knowledge, unique in offering efficient skipping. Large skipping capability allows to seed the generator for a large Monte-Carlo simulation on multiple clusters/CPU's with the absolute mathematical guarantee that the generated streams do not collide or overlap.

I would like to thank Jan Ambjorn, Fred James and George Savvidy, the three people whose encouragement is chiefly responsible for my persistence over the years with implementing the generator in software and studying its theoretical properties. I thank J. Apostolakis, J. Harvey and L. Moneta at CERN for useful discussions.

References

- [1] A.N. Kolmogorov, A new metrical invariant of transitive dynamical systems and automorphisms of Lebesgue spaces, Dokl. Acad. Nauk SSSR, vol. **119** (1958), p. 861
- [2] V.A. Rokhlin, On the endomorphisms of compact commutative groups, Izv. Akad. Nauk, vol. 13 (1949), p.329
On the entropy of automorphisms of compact commutative groups, Teor. Ver. i Pril., vol. 3, issue 3, (1961), p. 351
- [3] J. N. Franklin, Deterministic simulation of random processes, Math. Comp., 17 (1963), pp. 28-59.
Equidistribution of matrix-power residues modulo one, Math. Comp., 18 (1964), pp. 560-568.
- [4] D.V. Anosov, Geodesic flow on closed Riemannian manifolds of negative curvature, Nauka, Moscow (1967), in russian.
- [5] G. Savvidy and N. Ter-Arutyunyan-Savvidy, On the Monte Carlo simulation of physical systems, J.Comput.Phys. 97, 566 (1991)

- [6] N. Akopov, G. Savvidy and N. Ter-Arutyunyan-Savvidy, Matrix generator of pseudorandom numbers, *J.Comput.Phys.* 97, 573 (1991)
- [7] E.-H. A. D. E. Tahmi, Contribution aux générateurs de vecteurs pseudo-aléatoires, Thèse, Univ. Sci. Techn. Houari Boumedienne, Algiers, 1982.
- [8] N. Niki, Finite field arithmetic and multidimensional uniform pseudorandom numbers (in Japanese), *Proc. Inst. Statist. Math.* 32 (1984) 231.
- [9] H. Grothe, Matrix generators for pseudo-random vector generation, *Statist. Papers*, 28 (1987), pp. 233-238.
- [10] H.Niederreiter, A pseudorandom vector generator based on finite field arithmetic, *Mathematica Japonica*, Vol. 31, pp. 759-774, (1986)
- [11] G. G. Athanasiu, E. G. Floratos, G. K. Savvidy "K-system generator of pseudorandom numbers on Galois field," *International Journal of Modern Physics C*, Volume 8, Issue 03, pp. 555-565 (1997).
- [12] R. Lidl and H. Niederreiter, *Finite Fields*, Addison-Wesley, Reading, MA, 1983, see also "Finite fields, pseudorandom numbers, and quasirandom points," in : *Finite fields, Coding theory, and Advance in Communications and Computing*. (G.L.Mullen and P.J.S.Shine, eds) pp. 375-394, Marcel Dekker, N.Y. 1993.
- [13] G. Marsaglia and A. Zaman, *Ann. Appl. Probab.* Volume 1, Number 3 (1991), 462-480.
- [14] M. Luscher, *Computer Physics Communications* 79 (1994) 100
F. James, *Computer Physics Communications* 79 (1994) 111
- [15] M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator", *ACM Trans. on Modeling and Computer Simulation* Vol. 8, No. 1, January pp.3-30 (1998) DOI:10.1145/272991.272995
- [16] F. James, "Finally, a theory of random number generation." Fifth International Workshop on Mathematical Methods, CERN, Geneva, October 2001.
- [17] R.P. Brent and R. Zimmerman, Ten new primitive binary trinomials, *Math. Comp.* 78 (2009), 1197.
The Great Trinomial Hunt, *Notices of the AMS*, (2011)

- [18] G. Marsaglia, Random Numbers Fall Mainly in the Planes, Proc. Nat. Acad. Sci. 61,(1968), pp 25-28.
- [19] PARI/GP, version 2.5.5, Bordeaux, 2013, <http://pari.math.u-bordeaux.fr/>.
- [20] HEPFORGE.ORG, <http://mixmax.hepforge.org>
- [21] ROOT, <https://github.com/lmoneta/root/tree/random123>